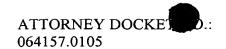2.   The system of Claim 1 wherein the plurality handler processes contain a plurality of threads, wherein each thread is operable to independently handle requests.

3.   The system of Claim 1 wherein the spawner process is operable to increase or decrease the number of handler processes currently in existence at any time, such operations known as load balancing.

4.   The system of Claim 1 wherein the server is composed of a plurality of physical processors, each processor operable to run one or more handler processes or the spawner process.

5.   **(Amended)** A method of operating a parallel client server system comprising:

creating a plurality of handler processes with a spawner process at a server;

initializing a well-known address at the server;

storing at least one request received by the well-known address in a buffer associated with the well-known address at the server;

notifying, in parallel, a plurality of the handler processes that at least one request has arrived;

accepting each pending request from the buffer, in parallel, with the plurality of handler processes when the number of handler processes exceeds the number of pending requests; and

accepting a number of pending requests substantially equal to the number of handler processes when the number of pending requests exceeds or equals the number of handler processes.

6.   The method of Claim 5 wherein attempting to accept pending requests from the buffer is also performed by a plurality of threads within the plurality of handler processes.

7.   The method of Claim 5 wherein creating the plurality of handler processes with the spawner process results in the plurality of processes running on a plurality of physical processors.

8.   **(Amended)** The method of Claim 5 and further comprising increasing or decreasing the number of handler processes currently in existence with the spawner process, such operations known as load balancing.

9.   The method of Claim 5 wherein the initialization of the well-known address is performed by cooperation between the operating system and the spawner process.

10.   Please cancel Claim 10 without prejudice or disclaimer.

11.   **(New)** A system of operating a parallel client server system comprising:

means for creating a plurality of handler processes with a spawner process at a server;

means for initializing a well-known address at the server;

means for storing at least one request received by the well-known address in a buffer associated with the well-known address at the server;

means for notifying, in parallel, a plurality of the handler processes that at least one request has arrived;

means for accepting each pending request from the buffer, in parallel, with the plurality of handler processes when the number of handler processes exceeds the number of pending requests; and

means for accepting a number of pending requests substantially equal to the number of handler processes when the number of pending requests exceeds or equals the number of handler processes.

12.   **(New)** A method of operating a parallel client server system comprising the steps of:

providing, at a server, at least one available handler process, the available handler process comprising a handler process which is not presently processing a previously accepted pending request;
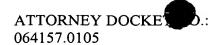
providing a well-known address at the server;

storing, at the server, at least one pending request received by the well-known address in a buffer associated with the well-known address;

notifying, substantially in parallel, the available handler processes that at least one pending request is in the buffer;

accepting substantially all pending requests from the buffer, substantially in parallel, with the available handler processes, when a number of pending requests is less than or equal to a number of available handler processes;

accepting a number of pending requests from the buffer substantially equal to the number of available handler processes when the number of pending requests is greater than the number of available handler processes; and
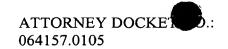
servicing accepted pending requests.

13.　(New)　The method of Claim 12 and further comprising processing error conditions with those available handler processes that did not successfully accept a pending request when the number of available handler processes is greater than the number of pending requests.

14.　(New)　The method of Claim 12, wherein providing the available handler processes comprises creating a plurality of the handler processes with a spawner process and wherein the available handler processes comprise a subset of the handler processes.

15.　(New)　The method of Claim 14, wherein notifying comprises updating a flag and wherein the flag is accessible by substantially all the handler processes at substantially any time.

16.　(New)　The method of Claim 12, wherein providing the well-known address comprises initializing the well-known address.

22.   (New) A system for a parallel client server system comprising:

means for providing, at a server, at least one available handler process, the available handler process comprising a handler process which is not presently processing a previously accepted pending request;

means for providing a well-known address at the server;

means for storing, at the server, at least one pending request received by the well-known address in a buffer associated with the well-known address;

means for notifying, substantially in parallel, the available handler processes that at least one pending request is in the buffer;

means for accepting substantially all pending requests from the buffer, substantially in parallel, with the available handler processes, when a number of pending requests is less than or equal to a number of available handler processes;

means for accepting a number of pending requests from the buffer substantially equal to the number of available handler processes when the number of pending requests is greater than the number of available handler processes; and

means for servicing accepted pending requests.

23.   (New) The method according to Claim 5 and further comprising:

servicing accepted requests with those handler processes that successfully accepted a pending request; and

processing error conditions with those handler processes that did not successfully accept a pending request.

17. **(New)** A system for a parallel client server system comprising:

software encoded on a computer readable medium, the software operable to:

provide, at a server, at least one available handler process, the available handler process comprising a handler process which is not presently processing a previously accepted pending request;

provide a well-known address at the server;

store, at the server, at least one pending request received by the well-known address in a buffer associated with the well-known address;

notify, substantially in parallel, the available handler processes that at least one pending request is in the buffer;

accept substantially all pending requests from the buffer, substantially in parallel, with the available handler processes, when a number of pending requests is less than or equal to a number of available handler processes;

accept a number of pending requests from the buffer substantially equal to the number of available handler processes when the number of pending requests is greater than the number of available handler processes; and

service accepted pending requests.

18. **(New)** The system of Claim 17, wherein the software is further operable to process error conditions associated with those available handler processes that did not successfully accept a pending request when the number of available handler processes is greater than the number of pending requests.

19. **(New)** The system of Claim 17, wherein the software is further operable to create a plurality of the handler processes with a spawner process and wherein the available handler processes comprise a subset of the handler processes.

20. **(New)** The system of Claim 19, wherein the software is further operable to update a flag associated with the notification and wherein the flag is accessible by substantially all the handler processes at substantially any time.

21. **(New)** The system of Claim 17, wherein the software is further operable to initialize the well-known address.